An improved threshold ring signature scheme based on error correcting codes

Pierre-Louis Cayrel, Sidi Mohamed El Yousfi Alaoui, Gerhard Hoffmann, Pascal Véron

An Improved Threshold Ring Signature Scheme Based on Error Correcting Codes

Pierre-Louis Cayrel[1], Sidi Mohamed El Yousfi Alaoui[2], Gerhrad Hoffmann[3], and Pascal Véron[4]

[1] Laboratoire Hubert Curien Université de Saint-Etienne, France
pierre-louis.cayrel@univ-st-etienne.fr
[2] CASED – Center for Advanced Security Research Darmstadt, Germany
elyousfi@cased.de
[3] Technische Universität Darmstadt, Germany
hoffmann@mathematik.tu-darmstadt.de
[4] IML/IMATH Université du Sud Toulon-Var. B.P.20132, F-83957 La Garde Cedex, France
veron@univ-tln.fr

Abstract. The concept of threshold ring signature in code-based cryptography was introduced by Aguilar et al. in [1]. Their proposal uses Stern's identification scheme as basis. In this paper we construct a novel threshold ring signature scheme built on the q-SD identification scheme recently proposed by Cayrel et al. in [14]. Our proposed scheme benefits of a performance gain as a result of the reduction in the soundness error from 2/3 for Stern's scheme to 1/2 per round for the q-SD scheme. Our threshold ring signature scheme uses random linear codes over the field $\mathbb{F}_q$, secure in the random oracle model and its security relies on the hardness of an error-correcting codes problem (namely the q-ary syndrome decoding problem). In this paper we also provide implementation results of the Aguilar et al. scheme and our proposal, this is the first efficient implementation of this type of code-based schemes.

Keywords: post-quantum cryptography, code-based cryptography, identification scheme, threshold ring signature scheme.

1 Introduction

The development in the field of quantum computing is a real threat to the security of many used public key cryptographic algorithms. Shor has demonstrated in 1994 that cryptographic schemes whose security relies on the difficulty of the factorization problem (e.g., RSA) and the difficulty of the discrete logarithm problem (e.g., DSA), could be broken using quantum computers. Consequently, it is necessary to have available alternative signature and identification schemes.

Coding based cryptography is one of the few alternatives supposed to be secure in a post quantum world. The most popular cryptosystems based on error-correcting codes are the McEliece [22] and Niederreiter [24] ones. The main

F. Özbudak and F. Rodríguez-Henríquez (Eds.): WAIFI 2012, LNCS 7369, pp. 45–63, 2012.

advantage of these two public key cryptosystems is the provision of a fast encryption and decryption procedure (about 50 times faster for encryption and 100 times faster for decryption than RSA).

Secure identification schemes were introduced by Feige, Fiat and Shamir [18]. These cryptographic schemes allow a prover to identify itself in polynomial time to a verifier without revealing any information of its secret key. Stern proposed at Crypto'93 [27] the first zero-knowledge identification scheme based on the syndrome decoding problem, this scheme could be turned into a digital signature via the Fiat-Shamir paradigm [17].

The concept of ring signatures was introduced first in 2001 by Rivest et al. [26]. Ring signatures permit any user from a set of intended signers to sign a message with no existing group manager and to convince the verifier that the author of the signature belongs to this set without revealing any information about its identity.

In 2002, Bresson et al. [12] extended ring signature schemes in a t-out-of-N threshold ring signature schemes, which enable any t participating users belonging to a set of N users to produce a signature in such a way that the verifier cannot determine the identity of the actual signers.

The concept of threshold ring signatures in code-based cryptography was introduced by Aguilar et al. in [1,2]. Their proposal is a generalization of Stern's identification scheme. The major advantage of this construction is that its complexity depends linearly on a maximum number of signers N, comparing with the complexity of threshold ring signature schemes based on number theory whose complexity is $\mathcal{O}(tN)$. However, the disadvantage of large public key size and signature length is still unsolved for this scheme.

Our Contribution: In this paper we propose an improved code-based threshold ring signature scheme. We achieve this by extending the five-pass zero-knowledge (q-SD) identification scheme proposed in [14] to a threshold ring identification scheme and applying the same idea as the Fiat-Shamir paradigm to transform it to a threshold ring signature. In this paper we provide also a first efficient implementation of the Aguilar et al. scheme and our scheme in order to show the advantage of our proposal in terms of performance.

Organization of the Paper: This paper is organized as follows: in Section 2 we recall some background on code-based cryptography. In Section 3 we present the q-SD identification scheme recently introduced by Cayrel et al. in [14], followed by some suggested improvements. We show in Section 4 how to use the q-SD identification scheme to construct our proposal, then we discuss the security and gives in detail the performance aspect of our construction by providing implementation results. Finally, we conclude the paper in Section 5.

2 Preliminaries

In this section, we recall useful notions of code-based cryptography. We refer to [9], for a general introduction to these issues.

2.1 Definitions

Linear codes are k-dimensional subspaces of an n-dimensional vector space over a finite field $\mathbb{F}_q$, where k and n are positive integers with $k < n$, and q a prime power. The error-correcting capability of such a code is the maximum number ω of errors that the code is able to decode. In short, linear codes with these parameters are denoted (n, k)-codes or $(n, n-r)$-codes, where r is the co-dimension of a code with $r = n - k$.

Definition 1 (Hamming weight). *The (Hamming) weight of a vector x is the number of non-zero entries. We use $\mathsf{wt}(x)$ to represent the Hamming weight of x.*

Definition 2 (Generator and Parity Check Matrix). *Let $\mathscr{C}$ be a linear code over $\mathbb{F}_q$. A generator matrix G of $\mathscr{C}$ is a matrix whose rows form a basis of $\mathscr{C}$:*

$$\mathscr{C} = \{xG : x \in \mathbb{F}_q^{n-r}\}$$

A parity check matrix H of $\mathscr{C}$ is defined by

$$\mathscr{C} = \{x \in \mathbb{F}_q^n : Hx^T = 0\}$$

and generates the dual space of $\mathscr{C}$.

Let n and r be two integers such that $n \geq r$, $\mathsf{Binary}(n, r)$ (resp. q-ary(n, r)) be the set of binary (resp. q-ary) matrices with n columns and r rows of rank r. Moreover, denote by $x \xleftarrow{\$} A$ the random choice of x among the elements of a set A.

We describe in the following the main hard problems on which the security of code-based schemes presented in this paper relies.

Definition 3 (Binary Syndrome Decoding (SD) problem).
Input : $H \xleftarrow{\$} \mathsf{Binary}(n, r)$, $y \xleftarrow{\$} \mathbb{F}_2^r$, and an integer $\omega > 0$.
Find : a word $s \in \mathbb{F}_2^n$ such that $\mathsf{wt}(s) \leq \omega$ and $Hs^T = y$.

This problem was proven to be NP-complete in 1978 [8]. It can be extended to arbitrary finite fields as follows.

Definition 4 (q-ary Syndrome Decoding (qSD) problem).
Input : $H \xleftarrow{\$} q\text{-}ary(n, r)$, $y \xleftarrow{\$} \mathbb{F}_q^r$, and an integer $\omega > 0$.
Find : a word $s \in \mathbb{F}_q^n$ such that $\mathsf{wt}(s) \leq \omega$ and $Hs^T = y$.

In 1994, A. Barg proved that this last problem remains NP-complete [5, in russian].

Definition 5 (q-ary Minimum Distance (qMD) problem).
Input : $H \xleftarrow{\$} q\text{-}ary(n, r)$, and an integer $\omega > 0$.
Find : a word $s \in \mathbb{F}_q^n$ such that $\mathsf{wt}(s) \leq \omega$ and $Hs^T = 0$.

Notice that the difficulties of solving the two problems (qSD and qMD) are equivalent [28]. The intractable assumptions associated to these problems are denoted by qSD assumption and qMD assumption, respectively.

Best known attack. The most efficient algorithm to attack code-based schemes is the Information Set Decoding (ISD) algorithm. Some improvements of this algorithm have been developed by Peters [25], Niebuhr et al. [23], and Bernstein et al. [10], and recently in [21] by Becker et al. and [7] by May et al.. The recent results of this attack are taken into account when choosing our parameters.

3 The q-SD Identification Scheme

An identification scheme is an interactive method for one party to prove to another that a statement is true, without revealing any additional information. In this section, we present the recent q-SD identification scheme based on error-correcting codes proposed by Cayrel et al. in [14]. The soundness error of approximately $1/2$ allows a performance gains when compared to Stern's scheme. For instance, one needs 16 rounds for the q-SD identification scheme and 28 rounds for the Stern's one to achieve the weak authentication probabilities of 2^{-16} according the norm ISO/IEC-9798-5.

3.1 Description of the q-SD Identification Scheme

In what follows, the elements of $\mathbb{F}_q^n$ are written as n blocks of size $\lceil \log_2(q) \rceil = m$ and each element of $\mathbb{F}_q$ is presented as m bits.

We first introduce a special transformation that will be used later.

Definition 6. *Let Σ be a permutation of $\{1, \ldots, n\}$ and $\gamma = (\gamma_1, \ldots, \gamma_n) \in \mathbb{F}_q^n$ such that $\forall i, \gamma_i \neq 0$. The transformation $\Pi_{\gamma,\Sigma}$ is defined as follows:*

$$\begin{array}{rccl} \Pi_{\gamma,\Sigma} : & \mathbb{F}_q^n & \longrightarrow & \mathbb{F}_q^n \\ & v & \mapsto & (\gamma_{\Sigma(1)} v_{\Sigma(1)}, \ldots, \gamma_{\Sigma(n)} v_{\Sigma(n)}) \end{array}$$

Notice that $\forall \alpha \in \mathbb{F}_q, \forall v \in \mathbb{F}_q^n, \Pi_{\gamma,\Sigma}(\alpha v) = \alpha \Pi_{\gamma,\Sigma}(v),$ and $\mathsf{wt}(\Pi_{\gamma,\Sigma}(v)) = \mathsf{wt}(v)$.

The q-SD identification scheme is comprised of two algorithms: key generation and identification protocol. Given the security parameter κ, the key generation algorithm described in Fig. 1 picks a random $(r \times n)$ q-ary matrix H common to all users, a random vector s (secret key) with Hamming weight ω. It outputs the public key y by multiplication of the vector s by a matrix H. The identification protocol described in Fig. 2 corresponds to a zero-knowledge proof of knowledge that the prover $\mathcal{P}$ possesses a private key s with a given public key y.

In Fig. 1, $\mathrm{WF}_{\mathrm{ISD}}$ denotes the Work-Factor of ISD algorithm over $\mathbb{F}_q$ and in Fig. 2, h denotes a hash function, S_n the symmetric group of degree n and $||$ the concatenation of two strings.

The authors of [14] proved that the protocol presented in Fig. 2 corresponds to a zero-knowledge interactive proof in the random oracle model, that means it satisfies the completeness, soundness, and zero-knowledge properties.

Key-Gen:
Choose n, r, ω, and q such that $\mathrm{WF}_{\mathrm{ISD}}(n, r, \omega, q) \geq 2^{\kappa}$
$H \xleftarrow{\$} \mathbb{F}_q^{r \times n}$
$s \xleftarrow{\$} \mathbb{F}_q^n$, s.t. $\mathsf{wt}(s) = \omega$.
$y \leftarrow Hs^T$
Output $(\mathrm{sk}, \mathrm{pk}) = (s, (y, H, \omega))$

Fig. 1. Key generation algorithm: parameters n, r, ω, q are public

Prover $\mathcal{P}(s,(y,H,\omega))$		**Verifier** $\mathcal{V}(y,H,\omega)$
$u \xleftarrow{\$} \mathbb{F}_q^n, \Sigma \xleftarrow{\$} S_n$		
$\gamma \xleftarrow{\$} \mathbb{F}_q^{n*}$		
$c_1 \leftarrow h\left(\Sigma \Vert \gamma \Vert Hu^T\right)$		
$c_2 \leftarrow h\left(\Pi_{\gamma,\Sigma}(u) \Vert \Pi_{\gamma,\Sigma}(s)\right)$	$\xrightarrow{c_1, c_2}$	
	$\xleftarrow{\alpha}$	$\alpha \xleftarrow{\$} \mathbb{F}_q$
$\beta \longleftarrow \Pi_{\gamma,\Sigma}(u + \alpha s)$	$\xrightarrow{\beta}$	
	$\xleftarrow{b}$	$b \xleftarrow{\$} \{0,1\}$
If $b = 0$:	$\xrightarrow{\Sigma, \gamma}$	**Check** $c_1 \stackrel{?}{=} h(\Sigma \Vert \gamma \Vert H\Pi_{\gamma,\Sigma}^{-1}(\beta)^T - \alpha y)$
Else:	$\xrightarrow{\Pi_{\gamma,\Sigma}(s)}$	**Check** $c_2 \stackrel{?}{=} h(\beta - \alpha\Pi_{\gamma,\Sigma}(s) \Vert \Pi_{\gamma,\Sigma}(s))$, $\mathsf{wt}(\Pi_{\gamma,\Sigma}(s)) \stackrel{?}{=} \omega$

Fig. 2. The q-SD identification protocol

Some Improvements of the q-SD Scheme:

- To get better communication complexity in comparison to the original version of the protocol presented in Fig. 2, the prover could use a public function φ_q^{-1} by sending $\varphi_q^{-1}(\Pi_{\gamma,\Sigma}(s))$ instead of $\Pi_{\gamma,\Sigma}(s)$, where φ_q is an efficient bijective encoding which takes its input from the interval $[0, (q-1)^{\omega}\binom{n}{\omega}[$ and outputs a binary word of length n and Hamming weight ω. This function is described in Algorithm 2 (see Appendix).
- We could use the same random seed to generate the permutation Σ and the vector γ in order to reduce more the communication complexity.

Proposed parameters. According to the ISD algorithm, the suggested parameters for the q-SD scheme are: $q = 256, n = 128, r = 64, \omega = \mathsf{wt}(s) = 49$.

Table 1 shows the advantage regarding the communication cost and the size of the public key of the q-SD scheme in comparison with Stern's initial proposal and his five-pass variant, for the same security level of 2^{80} and an impersonation

Table 1. Stern's schemes vs. the q-SD scheme, security level 2^{80}, probability of cheating 2^{-16}

	Stern (3-pass)	Stern (5-pass)	q-SD
Rounds	28	16	16
Public data (bits)	122850	124950	33280
Secret key (bits)	700	4900	1024
Communication complexity (bits)	42019	62272	39056
Prover's Computation	$2^{22.7}$op. over $\mathbb{F}_2$	$2^{21.92}$op. over $\mathbb{F}_2$	2^{16}mult + 2^{16}add op. over $\mathbb{F}_{256}$

resistance of 2^{-16}. It is considered that all seeds used are 128 bits long and that all hash values are 160 bits long.

Remark 1. To be fair, the two improvements above-mentioned according the scheme q-SD are not taken into account in the calculation of the communication complexity in table 1, since these improvements can be applied to Stern's schemes. However by using the same seed for Σ and γ, the communication complexity will be 30864 bits, and only 26760 bits if we use in addition an encoding function.

Remark 2. We want to mention that the authors in [3] propose a new five-pass identification scheme with small size of keys and an asymptotic cheating probability of 1/2, this scheme is related to the Véron identification scheme and its security is based on the syndrome decoding Problem. The idea of this construction is based on deriving new challenges from the secret key through cyclic shifts of the initial public key syndrome.

3.2 Signature Schemes from Identification Schemes

One efficient method to derive a signature from an identification scheme is given via the Fiat-Shamir paradigm [17]. Pointcheval and Stern use the well-known forking lemma, in order to provide the security argument of signatures obtained from three-pass identification protocols. The authors in [4] extend the Fiat-Shamir transform and the Forking lemma to obtain secure signatures from identification protocols with multi-pass. For instance, in the case of a five-pass identification scheme, the signer replaces the two moves given from the verifier by the outputs of some random oracles, he sets the transcript $(\sigma_1||h_1||\sigma_2||h_2||\sigma_3)$ as a signature of a message M, where h_1 and h_2 the outputs of two hash functions $\mathcal{H}_1$ respectively $\mathcal{H}_2$ modeled as random oracles, and the σ_1, σ_2 and σ_3 the values given from the prover as in the identification scheme. Similarly to [20], the authors of [4] generalize the forking lemma even more for ring signatures schemes in order to give security arguments for such a class of signature.

4 Code-Based Threshold Ring Signature Schemes

The notion of threshold ring signature is introduced in 2002 by Bresson et al. [12]. We first define the formal definition of the threshold ring identification scheme which can be turned to get a threshold ring signature in accord with Section 3.2.

Definition 7. *Let $t < N$ be integers. We assume that each user P_i owns the pair of keys (sk_i, pk_i) corresponding respectively to the secret and the public keys. Let $P_1, \ldots, P_N$ be the N potential provers of the ring with their public keys $pk_1, \ldots, pk_N$. Let t of the N members form a group of provers, one of them is the leader L. The threshold identification scheme consists of the algorithms:*

- ***Setup*** *takes a secret parameter as input and outputs the public parameters and chooses the leader.*
- ***Ring key generation*** *takes public parameters as input and outputs a pair of keys corresponding to the secret and the public key.*
- ***Commitment-challenge-answer and verification step*** *is an interactive protocol between the t-users and the verifier consisting of the computation of the commitments, challenges and responses, following by a verification step which takes as input the answers of the challenges and verifies the honestly of the computation, and returns 1 (accept), and 0 (reject).*

For the security of a threshold ring signature scheme the basic criteria are:

- Unforgeablity: Without the knowledge of the t secret keys, it is infeasible to generate a valid (t, N) threshold ring signature.
- Anonymity: Given a message-signature pair, it should be infeasible for the verifier to reveal which t-subset of signers generated a signature.

See defintion 3 and 4 in [2] for a formal definition of the two above properties. In [1], Aguilar et al. introduced the first code-based threshold ring signature. The main idea of this scheme is to generalize the Stern's identification scheme and then convert this latter into a threshold ring signature using the Fiat-Shamir paradigm. Aguilar et al.'s scheme is proven to be a zero-knowledge protocol with soundness error of 2/3 as in the Stern's protocol for each round. Its security relies on the hardness of the binary Minimum Distance problem.

A second code-based threshold ring signature scheme has been proposed by Dallot and Vergnaud in [16]. Their proposal is not derived from an identification scheme. It uses Goppa codes and combines the generic construction of Bresson et al. [12] and the CFS signature scheme [15]. The authors of [16] obtained a short signature but the required time to generate it, is too high, and the huge public key size is also a disadvantage of their proposal.

Using the Aguilar et al.'s approach, Cayrel et al. proposed in [13] a lattice-based threshold ring signature scheme based on the hardness of the SIS problem.

The soundness error of approximately 1/2 for the q-SD identification scheme allows a performance gains when compared to Stern's scheme. In order to make use of this gain, we present in this section a novel threshold ring identification scheme and according to Section 3.2, we turn it into a threshold ring signature scheme.

To describe our scheme, we need the two notions of block permutations:

Definition 8. *Let n and N be two integers and let $\beta = (\beta_1, \ldots, \beta_n, \beta_{n+1}, \ldots, \beta_{2n}, \ldots, \beta_{nN})$ be a vector of length nN defined over*

some alphabet. Let us define for $i \in [1, N]$ *the elements* $\tilde{\beta}_i = (\beta_{(i-1)n+1}, \ldots, \beta_{in})$ *such that* β *can be expressed as* $(\tilde{\beta_1}, \ldots, \tilde{\beta_N})$.

The constant (n, N)*-block permutation* Θ *is a permutation over* $\{1, \ldots, N\}$ *which acts over vectors of length* nN *such that*

$$\Theta(\beta) = \Theta(\tilde{\beta_1}, \ldots, \tilde{\beta_N}) = (\tilde{\beta}_{\Theta(1)}, \ldots, \tilde{\beta}_{\Theta(N)})$$

Let $\sigma = (\sigma_1, \ldots, \sigma_N)$ *be a family of* N *permutations over* $\{1, \ldots, n\}$*, we define a* (n, N)*-block permutation* Π*, as a permutation which acts over a vector of length* nN *and which is the product of a constant* n*-block permutation* Θ *and the family* σ*, i.e.*

$$\Pi(\beta) = \Theta(\sigma_1(\tilde{\beta_1}), \ldots, \sigma_N(\tilde{\beta_N}))$$

Informally speaking a constant (n, N)*-block permutation divides a vector of length* nN *into* N *blocks of size* n *and permutes them. A* (n, N)*-block permutation permutes also for each block the components of the block.*

Example 1. The permutation $(6, 5, 4, 3, 2, 1)$ is $(2, 3)$-block permutation, and the permutation
$(3, 4, 5, 6, 1, 2)$ is a constant $(2, 3)$-block permutation since the order on each block $((1, 2), (3, 4)$ and $(5, 6))$ is preserved in the block permutation.

4.1 Description of Our Threshold Identification Protocol

We consider one set of N members. Let $(P_1, \ldots, P_t)$ be a subset of this set consisting of the members who want to prove that they know some secret s, whereas one of them is a leader L. The parameter t corresponding to the number of provers has to be fixed at the beginning of the protocol.

Our protocol consists of the following steps: Setup, Ring public key generation, Commitment-Challenge-Answer and Verification step. We can formally describe each step as follows:

- ***Setup*** Given κ as security parameter, we generate the corresponding public parameters n, r, ω, and q such that $\mathrm{WF}_{\mathrm{ISD}}(n, r, \omega, q) \geq 2^{\kappa}$, where n and r are the parameters for each matrix H_i $(1 \leq i \leq N)$ which will be used to form the ring public matrix. Each matrix can be constructed as follows: we choose a random vector $s_i \in \mathbb{F}_q^n$ of weight ω, generate $n - r - 1$ random vectors and consider the code $\mathscr{C}_i$ obtained by these $n - r$ words (the operation can be repeated until the co-dimension of $\mathscr{C}_i$ is r). The matrix H_i is then a parity-check matrix of a code $\mathscr{C}_i$ and thus we have $H_i s_i^T = 0$, where $s_i \in \mathbb{F}_q^n$ has a weight ω. The fact that we take a same syndrome and the same weight for the vectors s_i helps for conserving the anonymity in the group. For the $(N - t)$ other users, s_i are fixed at 0, because 0 is always a solution of the equation $H_i s_i^T = 0$.

- ***Ring key generation*** The leader collects all these matrices and forms among them a public key $(H, t\omega)$ called ring public key, the matrix H can be described as follows:

$$H = \begin{pmatrix} H_1 & 0 & \cdots & 0 \\ 0 & H_2 & 0 & 0 \\ \vdots & \ddots & H_i & 0 \\ 0 & 0 & \cdots & H_N \end{pmatrix}$$

– ***Commitment-challenge-answer and verification step*** To simplify the description, we consider that the t provers correspond to the first matrices H_i $(1 \leq i \leq t)$. The leader L, member of the set of t provers among N members, want to prove to the verifier that he knows a secret key s, where s is a nN vector of weight tw. This will be achieved by performing the following steps:
  - Each member of the t provers (including L) creates local commitments using the secret keys s_i and sends them to L.
  - L collects all these commitments, simulates the missing ones for the $(N-t)$ other users by fixing all remaining s_i by 0, and create the master commitment using a random constant block permutation.
  - The master commitment are sent to the verifier V.
  - V chooses a random value α over $\mathbb{F}_q$ and sends it to L, the latter one forwards this value to the $(t-1)$ provers.
  - Each member of the t provers (including L) calculates the vectors β_i, L collects those values and creates a global vector β' using a constant block permutation and it will be sent to V.
  - V chooses a challenge from $\{0, 1\}$ and sends it to L who forwards it to the $(t-1)$ provers.
  - L collects the answers from the $(t-1)$ provers, computes the responses for the other users and finally computes a global answer for V.
  - After receiving the global answer, V checks the correctness of the master commitments.

Algorithm 1 gives a full description of this interaction between the set of t provers and the verifier. This algorithm has to be performed in multi-rounds in order to reach the required impersonation resistance.

We stress that during the answer step (line 19 of Algorithm 1), the knowledge of the permutation ρ permits to recover Θ, Σ_i, and γ_i for $1 \leq i \leq N$. In addition, the verifier can easily obtain $\beta_i (1 \leq i \leq N)$ by applying the inverse of θ on the known vector β'.

4.2 Security

We first prove that the generalized q-SD identification protocol is an honest-verifier zero-knowledge proof of knowledge. The resulting threshold ring signature obtained from the application of the work presented in [4] on the generalized q-SD identification protocol is existentially unforgeable under chosen message attacks in the random oracle model.

Lemma 1. *Finding a vector s of length nN such that the global weight of s is tw, the weight of s for each of the N blocks of length n is 0 or ω, and such that s has a null syndrome for H, is hard under the assumption of hardness of the qMD problem.*

Algorithm 1. Generalized q-SD protocol

INPUT: $n, k, N, t \in \mathbb{N}$, where $k < n$ and $t < N$.
$H \in \mathbb{F}_q^{rN \times nN}$, where $r = n - k$ and h a collision resistant hash function.

PRIVATE KEY: $s = (s_1, \ldots, s_N) \in \mathbb{F}_q^{nN}$, $\mathsf{wt}(s_j) = 0$ or $\mathsf{wt}(s_j) = \omega$ with $\mathsf{wt}(s) = t\omega$ and $Hs^T = 0$.

COMMITMENT STEP:

Each prover P_i chooses $u_i \xleftarrow{\$} \mathbb{F}_q^n$, $\Sigma_i \xleftarrow{\$} S_n$, $\gamma_i \xleftarrow{\$} \mathbb{F}_q^{n*}$ $(1 \leq i \leq t)$.
P_i constructs $c_{1,i} \leftarrow h\left(\Sigma_i || \gamma_i || H_i u_i^T\right)$ and $c_{2,i} \leftarrow h\left(\Pi_{\gamma_i, \Sigma_i}(u_i) || \Pi_{\gamma_i, \Sigma_i}(s_i)\right)$.
P_i sends $c_{1,i}$ and $c_{2,i}$ to leader L.
L fixes the secret keys s_i of the $N - t$ other users at 0 $(t + 1 \leq i \leq N)$.
L chooses $N-t$ values $u_i \xleftarrow{\$} \mathbb{F}_q^n$ and $N-t$ permutations $\Sigma_i \xleftarrow{\$} S_n$ and $N-t$ values $\gamma_i \xleftarrow{\$} \mathbb{F}_q^{n*}$ $(t+1 \leq i \leq N)$.
L chooses $\Theta \xleftarrow{\$} S_N$ in order to obtain the master commitments.
L computes the master commitments $C_1 \leftarrow h(\Theta || c_{1,1} || \ldots || c_{1,N})$ and $C_2 \leftarrow h(\Theta(c_{2,1}, \ldots, c_{2,N}))$.
C_1 and C_2 are sent to the verifier V.
V sends back the value $\alpha \xleftarrow{\$} \mathbb{F}_q$ and L passes it to each P_i $(1 \leq i \leq t)$.
P_i computes $\beta_i \longleftarrow \Pi_{\gamma_i, \Sigma_i}(u_i + \alpha s_i)$ $(1 \leq i \leq t)$.
L computes $\beta_i \longleftarrow \Pi_{\gamma_i, \Sigma_i}(u_i)$ $(t + 1 \leq i \leq N)$.
$\beta' = \Theta(\beta) = \Theta(\beta_1, \cdots, \beta_N) = (\beta_{\Theta(1)}, \ldots, \beta_{\Theta(N)})$ is sent to V.

CHALLENGE STEP:

V sends a challenge $b \xleftarrow{\$} \{0, 1\}$

ANSWER STEP: ▷ The first part of this step is between each prover P_i $(1 \leq i \leq t)$ and the leader L.

if $b = 0$ **then**
  P_i sends γ_i and Σ_i to L.
else if $b = 1$ **then**
  P_i sends $\varphi_q^{-1}(\Pi_{\gamma_i, \Sigma_i}(s_i))$ to L.
end if ▷ φ_q^{-1} is an implementation detail not necessary for the algorithm.

L simulates the $N - t$ other answers with $s_i = 0$ $(t + 1 \leq i \leq N)$.
L computes the answer for V:

if $b = 0$ **then**
  $\gamma = (\gamma_1, \ldots, \gamma_N)$, $\Sigma = (\Sigma_1, \ldots \Sigma_N)$, and Θ are sent to V.
else if $b = 1$ **then**
  $\rho(s) = (\Pi_{\gamma_{\Theta(1)}, \Sigma_{\Theta(1)}}(s_{\Theta(1)}), \ldots, \Pi_{\gamma_{\Theta(N)}, \Sigma_{\Theta(N)}}(s_{\Theta(N)}))$ is sent to V. ▷ NB: Not Θ.
end if

VERIFICATION STEP:

if $b = 0$ **then**
  V checks $C_1 \stackrel{?}{=} h(\Theta || h(\Sigma_1 || \gamma_1 || H_1 \Pi^{-1}_{\gamma_1, \Sigma_1}(\beta_1)^T) || \cdots || h(\Sigma_N || \gamma_N || H_N \Pi^{-1}_{\gamma_N, \Sigma_N}(\beta_N)^T))$ and $\Theta \stackrel{?}{\in} S_N$.
else if $b = 1$ **then**
  V checks

$$C_2 \stackrel{?}{=} h\left(\begin{pmatrix} h(\beta_{\Theta(1)} - \alpha \Pi_{\gamma_{\Theta(1)}, \Sigma_{\Theta(1)}}(s_{\Theta(1)}) \,||\, \Pi_{\gamma_{\Theta(1)}, \Sigma_{\Theta(1)}}(s_{\Theta(1)})) \\ h(\beta_{\Theta(2)} - \alpha \Pi_{\gamma_{\Theta(2)}, \Sigma_{\Theta(2)}}(s_{\Theta(2)}) \,||\, \Pi_{\gamma_{\Theta(2)}, \Sigma_{\Theta(2)}}(s_{\Theta(2)})) \\ \vdots \\ h(\beta_{\Theta(N)} - \alpha \Pi_{\gamma_{\Theta(N)}, \Sigma_{\Theta(N)}}(s_{\Theta(N)}) \,||\, \Pi_{\gamma_{\Theta(N)}, \Sigma_{\Theta(N)}}(s_{\Theta(N)})) \end{pmatrix}^T\right),$$

  $\mathsf{wt}(\rho(s)) \stackrel{?}{=} t\omega$, and that $\rho(s)$ is formed of N blocks of length n and of weight ω or weight 0.
end if

Proof. The construction of the matrix H (described above) and the vector s implies that finding such a n-block of length nN is also equivalent to finding a solution of a local hard problem s_i of weight ω such that $H_i s_i = 0$, which is hard under our assumption.

Theorem 1. *Our scheme is an honest verifier zero-knowledge proof of knowledge, with soundness error bounded by* $1/2$*, that the group of* t *provers knows a vector* s *of length* nN *such that the global weight of* s *is* $t\omega$*, and such that the vector* s *has a null syndrome for* H*. The scheme is secure in the random oracle model under the assumption of the hardness of the qMD problem.*

Proof. We prove that our scheme satisfies the three properties: completeness, soundness and zero-knowledge.

Completeness: It is clear that each group of honest provers who has the knowledge of a valid secret key is able to answer correctly any of the honest leader's queries, which permit him to compute the master commitments. The leader, on his turn is able to reveal the information necessary to the honest verifier, in order to check the correctness of these commitments.

Soundness: In [14] it was proven that the regular q-SD scheme satisfies this property and that the soundness error is bounded by $1/2$, assuming that the qMD problem is hard. Because our protocol can be seen as a composition of t simultaneous executions of the q-SD scheme and given that the latter one can be reduced to our protocol by making all signing instances equal, this implies that this soundness error cannot be higher than $1/2$ for our protocol in one single round.

Zero-knowledge: The zero-knowledge property for our protocol can be proven in the random oracle model. In order to do that, we use the classical idea of resettable simulation. Let M be a polynomial-time probabilistic turing machine (simulator) using a dishonest verifier. Because of the two interactions with the leader (prover for our case), we have to assume that the dishonest verifier could contrive two strategies: $St_1(C_1, C_2)$ taking as input the leader's (master) commitments and generating a value $\alpha \in \mathbb{F}_q$, $St_2(C_1, C_2, \beta')$ taking as input the leader's commitments, the answer β and generating as output a challenge in the set $\{0, 1\}$. M will generate a communication tape representing the interaction between leader and verifier. The goal is to produce a communication tape whose distribution is indistinguishable from a real tape by an honest interaction. The simulator M is constructed as follows:

Step 1. M randomly picks a query b from $\{0, 1\}$.

- If $b = 0$, M randomly chooses: u_i, γ_i, Σ_i $(1 \leq i \leq N)$ and Θ as a random constant block permutation on N blocks $\{1, 2, \ldots, N\}$, and solves the equation: $H{s'}^T = y$ for some vector $s' = (s'_1, \ldots, s'_N)$ of length nN and not necessarily satisfying the condition $\mathsf{wt}(s') = t\omega$. The values $c_{1,i}$ $(1 \leq i \leq N)$ can be computed as follows: $c_{1,i} = h\left(\Sigma_i || \gamma_i || H_i u_i^T\right)$, the master commitments are taken then as $C_1 = h(\Theta || c_{1,1} || \ldots || c_{1,N})$ and C_2 as a random string. By simulating the verifier, M applies $St_1(C_1, C_2)$ to get $\alpha \in \mathbb{F}_q$, and then computes β' as follows: $\beta' = \Theta(\Pi_{\gamma_1, \Sigma_1}(u_1 + \alpha s'_1), \ldots, \Pi_{\gamma_N, \Sigma_N}(u_N + \alpha s'_N))$,

and has the information needed to derive the simulated communication data between leader and verifier. Therefore the candidates to be written in the communication tape consist of elements $A = C_1||C_2$, β' and $ans = \rho = \Theta(\Pi_{\gamma_1,\Sigma_1}, \ldots, \Pi_{\gamma_N,\Sigma_N})$. Taking into account the uniform distribution of the random variables used in the computation of A, ans and β', it follows that the distribution of these elements is indistinguishable from those resulting from a fair interaction.

– If $b = 1$, M randomly chooses u_i, γ_i, Σ_i $(1 \leq i \leq N)$ and Θ as a random constant block permutation on N blocks $\{1, 2, \ldots, N\}$. This time it picks $s = (s_1, \ldots, s_N)$ as a random vector from the set $\mathbb{F}_q^{nN}$ with weight $t\omega$ and formed of N blocks of length n and of weight ω or 0. The commitments C_1 will be given uniformly at random values and $C_2 = h(\Theta(c_{2,1}, \ldots, c_{2,N}))$ such that each $c_{2,i} = h(\Pi_{\gamma_i,\Sigma_i}(u_i)||\Pi_{\gamma_i,\Sigma_i}(s_i))$. Again, from $St_1(C_1, C_2)$, M gets $\alpha \in \mathbb{F}_q$ and computes β' as follows: $\beta' = \Theta(\Pi_{\gamma_1,\Sigma_1}(u_1 + \alpha s_1), \ldots, \Pi_{\gamma_N,\Sigma_N}(u_N + \alpha s_N))$, and has the information needed to derive the simulated communication data. The communication set features elements $A = C_1||C_2$, β' and $ans = \rho(s) = (\Pi_{\gamma_{\Theta(1)},\Sigma_{\Theta(1)}}(s_{\Theta(1)}), \ldots, \Pi_{\gamma_{\Theta(N)},\Sigma_{\Theta(N)}}(s_{\Theta(N)}))$. The uniformly random character of the choices made will render these elements indistinguishable from those resulting from a fair interaction.

Step 2. M applies the verifier's strategy obtaining b' as result.

Step 3. When $b = b'$, the machine M writes on its communication tape the values of A, α, β', b and ans. If the values differ, however, nothing is written and the machine returns to Step 1.

Therefore, in 2δ rounds on average, M produces a communication tape indistinguishable from one that corresponds to a fair interaction process execution that takes δ rounds. □

We derive the following corollary due to the results from [4].

Corollary 1. *The resulting threshold ring signature scheme obtained from the honest verifier zero-knowledge q-SD identification protocol is unforgeable under chosen message attacks in the random oracle model.*

Theorem 2. *Our threshold ring signature scheme obtained from the q-SD identification protocol is anonymous in the random oracle model.*

Proof. The second property we would like to examine is the anonymity property in the random oracle model of the resulting threshold ring signature scheme obtained from our q-SD identification protocol. In other words, the verifier must not be able to determine the identity of the real signers, a part from the fact that they were at least t among the n specified ring members. For the challenge 0 the response of both real signers and the non-signers are completely indistinguishable, since Θ, Σ_i, and γ_i are chosen uniformly at random and therefore the response is random. So the only possibility to identify non-signers is challenge 1. In this case the verifier receives a permuted value of the secret key without having access to the used permutation. As consequence, the anonymity of the signers is preserved. □

Table 2. Comparison code-based threshold ring signature schemes

Threshold ring Signatures	Pk size in KBytes	Sign. size in KBytes	Sign. cost in bops
Aguilar et. al's scheme	1470	2448	2^{30}
Dallot and Vergnaud's scheme	10137122	7	2^{35}
Our scheme	400	2384	2^{26}

4.3 Performance and Implementation

Theoretical Results. In general the signature length of signature schemes derived from identification schemes is constrained by the number of rounds. Our proposal is built by applying the q-SD identification scheme, which needs a smaller number of rounds to reach the same probability of cheating as Stern's identification scheme. For a probability of cheating of 2^{-80}, one needs about 140 rounds for Stern's identification scheme and only 80 rounds for our proposal. This fact has a positive effect in terms of signature length for our proposal as we will see below.

Dallot and Vergnaud's scheme uses a CFS signature scheme as basis, therefore it inherits the advantage to provide a shorter signature length, however it suffers from slow signature generation cost and large key sizes compared to our construction.

In Table 2 we give our results in terms of key sizes, signature length and the signing cost for the parameter set $(N, t) = (100, 50)$ in comparison with the code-based threshold ring signature schemes due to Aguilar et al. [1] and Dallot and Vergnaud [16].

Before we do that we have to mention that the signature length of our scheme is N times the signature length of one q-SD signature length, similarly is for the computation of the public key size and the complexity of the protocol.

For our scheme, taking into account the performances of the ISD algorithm, we suggest the following parameters: $q = 256$, $n = 128$, $r = 64$, and $\omega = 49$.

For the same security level, we need to take respectively: $q = 2$, $n = 694$, $r = 347$, $\omega = 69$ for Aguilar et al.'s scheme [1] and, $q = 2$, $n = 2^{22}$, $r = 198$, $\omega = 9$ for Dallot and Vergnaud's scheme [16].

We considered that all seeds used are 128 bits long, the hash outputs are 160 bits long, the security level is 2^{80}, and that the probability of cheating is bounded by 2^{-80}.

Table 2 shows that for the same level of security, the public key size for our scheme is almost four times smaller than Aguilar et. al's one and 25344 times smaller than Dallot and Vergnaud's one. The signature length of our construction is 2384 Kbytes, where that of Aguilar et al.'s scheme is 2448 Kbytes and almost 7 Kbytes for Dallot and Vergnaud's one. For signing cost, we obtain better results.

Using the two improvement presented in Section 3, the signature length of our threshold ring signature scheme is almost 1633 KBytes.

We also compare our scheme to the recent work [13], which is a lattice based threshold ring signature based on the hardness of SIS problem and is related to this work.

For a 111 bit-security, the authors of [13] obtain 45 Mbytes for the signature length. Using our scheme the signature length is only 4 Mbytes for the same security level and the same hash length (224 bits), using parameters $q = 256, n = 204, r = 102$, and $\omega = 71$.

Remark 3. To reduce more the public key size, we can use the proposal in [19] and [6] by replacing a random matrix H by a double circulant matrix respectively a quasi-dyadic matrix. In this case, we obtain a public key size in 12.5 Kbytes for our construction, 8.47 Kbytes for Aguilar et al.'s one. and 1146 Kbytes for Dallot and Vergnaud's scheme.

4.4 Practical Results

General Remarks. The following tables show the timings we have obtained for a C implementation. The test system was an Intel(R) Core(TM)2 Duo CPU E8400@3.00GHz, running Debian 6.0.3. The sources have been compiled using gcc 4.6.2.

In all cases, we used parity check matrices in systematic form. Due to the row-major order of C, the transposed matrices have been stored. The number of the ring members was always set to 100, the number of ring provers to 50. The tables show the setup time and the time running the protocol, where the setup time is consumed for the generation of the necessary public and private keys.

Finally, the use of quasi-dyadic matrices does not allow for all theoretically possible parameters. For instance, dyadic matrices have the dimension $2^p \times 2^p$ ($p \in \mathbb{N}$), which means that for quasi-dyadic matrices $r = d2^p$ for some $d \in \mathbb{N}$. In order to have comparable results and a uniform implementation, we have used this restriction for the random and the quasi-cyclic case as well.

Aguilar et. al Scheme. The number of rounds for the scheme has been set to 28 (probability of cheating 2^{-16}), the dimension of the parity check matrix H^T over $\mathbb{F}_2$ has been set to 704×352, but only the redundancy part has been stored in memory, which is of dimension 352×352. The weight of the secrets been set to 76 (Table 3).

Our Scheme. For our scheme the parity check matrices H^T have been chosen over $\mathbb{F}_{2^8}$, mainly because in this case a field element fits exactly in one byte. The

Table 3. Aguilar et al. timings for 28 protocol rounds, $H \in \mathbb{F}_2^{352 \times 704}$

Matrix Type	**Dim.** $[n \times r]$	**Weight**	**Setup** $[ms]$	**Protocol** $[ms]$	**Total** $[ms]$	**Sec**
Random	704×352	76	108.539	98.662	207.200	2^{80}
Quasi-dyadic	704×352	76	811.202	474.737	1285.939	2^{80}
Quasi-cyclic	704×352	76	476.796	302.935	779.731	2^{80}

Table 4. Timings for our scheme and 16 protocol rounds, $H \in \mathbb{F}_{2^8}^{72 \times 144}$

Matrix Type	**Dim.** $[n \times r]$	**Weight**	**Setup** $[ms]$	**Protocol** $[ms]$	**Total** $[ms]$	**Sec**
Random	144×72	54	32.979	18.499	51.477	2^{80}
Quasi-dyadic	144×72	54	44.331	29.109	73.439	2^{80}
Quasi-cyclic	144×72	54	38.747	26.550	65.298	2^{80}

number of rounds has been set to 16 (probability of cheating 2^{-16}), the weight of the secrets has been set to 54. The number of the ring members was again set to 100 and the number of ring provers to 50 (Table 4).

Remark 4. As one can see, the computational cost for quasi-dyadic/-cyclic cases is always higher than using random parity check matrices. The reason is that the vector-matrix product is more expensive in those cases, because the matrix has to be reconstructed on the fly during the multiplication without actually building the whole matrix in memory. The savings in memory have to be paid for with additional runtime.

Remark 5. The given implementation is given as a proof of concept. For instance, the communication between the leader and the provers takes place on the same machine, even inside the same executable. In reality, the provers would be located on different computers, having a different architecture, connected to the leader via network connections and the like. In such a heterogeneous scenario, the communication latency for those network connections had to be taken into account. It also might be possible that some provers use a very fast machine, whereas others use a very slow one. The interaction process would be dominated then by the slowest possible prover.

Transforming into a Signature Scheme. Similar to the general technique shown by Fiat-Shamir, we can transform our scheme into a signature scheme. The idea is that the signing and verifying part are handled separately, i.e. at first, the leader simulates the challenge step of the verifier using a public stream cipher or public hash function with some predefined starting value involving the document to sign. The protocol is then run without further verification, but all the data which are needed for verification, in particular the master commitments, has to be recorded. These data form the signature.

The verifier uses the recorded data to run the protocol without the signing side. For the challenge step, the verifier uses the same starting value for the stream cipher or hash function as the signing part did. The document is part of this starting value. A consequence of this approach is that the signatures become quite large as everything needed for verification has to be recorded in the signature.

In Table 5 we give some timings for the resulting signature scheme. We used the same settings as above, but run the protocol with random matrices only. The savings using other matrix types is negligible compared to the gained signature sizes.

Table 5. Timings for 80 protocol rounds, $H \in \mathbb{F}_{2^8}^{72 \times 144}$

Doc. $[MiB]$	**Sig.** $[MiB]$	**Dim.** $[n \times r]$	**Weight**	**Signing** $[ms]$	**Verification** $[ms]$	**Total** $[ms]$	**Sec**
1	4	144×72	54	544	454	998	2^{80}
10	13	144×72	54	3643	3551	7194	2^{80}
25	28	144×72	54	8803	8700	17503	2^{80}

The signature sizes are not fixed, but show a small variation depending on the values chosen during the challenge step. More specifically, the answers transmitted for the cases b=0,1,2 vary in size, which effectively leads to varying signature sizes as well. The values are therefore average values obtained while running the protocol 80 rounds (probability of cheating 2^{-80}).

5 Conclusion

Starting from the recently proposed q-SD zero-knowledge identification scheme [14], we presented in this work a novel threshold ring signature scheme based on error-correcting codes based on the q-SD identification scheme. Since the q-SD scheme has a low soundness error allowing a specified security to be reached in few rounds, our construction uses this fact to achieve a scheme which shorter signature length, smaller public key size and signature cost compared to Aguilar et al.'s which is based on Stern's identification scheme. We have confirmed our results by implementing the both schemes in C that shows clearly the advantage of our proposal. The source code of our implementation can be found here: http://www.cayrel.net/IMG/tgz/waifi-files.tgz.

References

1. Aguilar Melchor, C., Cayrel, P.-L., Gaborit, P.: A New Efficient Threshold Ring Signature Scheme Based on Coding Theory. In: Buchmann, J., Ding, J. (eds.) PQCrypto 2008. LNCS, vol. 5299, pp. 1–16. Springer, Heidelberg (2008)
2. Aguilar Melchor, C., Cayrel, P.-L., Gaborit, P., Laguillaumie, F.: A new efficient threshold ring signature scheme based on coding theory. IEEE Transactions on Information Theory 57(7), 4833–4842 (2011)
3. Aguilar Melchor, C., Gaborit, P., Schrek, J.: A new zero-knowledge code based identification scheme with reduced communication (2011), `http://arxiv.org/PS_cache/arxiv/pdf/1111/1111.1644v1.pdf`
4. El Yousfi Alaoui, S.-M., Dagdelen, Ö., Véron, P., Galindo, D., Cayrel, P.-L.: Extended security arguments for (ring) signature schemes. Cryptology ePrint Archive, Report 2012/068 (2012)
5. Barg, S.: Some new NP-complete coding problems. Problemy Peredachi Informatsii 30, 23–28 (1994)
6. Barreto, P.S.L.M., Cayrel, P.-L., Misoczki, R., Niebuhr, R.: Quasi-Dyadic CFS Signatures. In: Lai, X., Yung, M., Lin, D. (eds.) Inscrypt 2010. LNCS, vol. 6584, pp. 336–349. Springer, Heidelberg (2011)

7. Becker, A., Joux, A., May, A., Meurer, A.: Decoding Random Binary Linear Codes in $2^{(n/20)}$: How $1 + 1 = 0$ Improves Information Set Decoding. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 520–536. Springer, Heidelberg (2012)
8. Berlekamp, E., McEliece, R., van Tilborg, H.: On the inherent intractability of certain coding problems. IEEE Transactions on Information Theory 24(3), 384–386 (1978)
9. Bernstein, D.J., Buchmann, J., Dahmen, E.: Post-Quantum Cryptography, 1st edn. Springer Publishing Company, Incorporated (2008)
10. Bernstein, D.J., Lange, T., Peters, C.: Smaller Decoding Exponents: Ball-Collision Decoding. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 743–760. Springer, Heidelberg (2011)
11. Biswas, B., Sendrier, N.: McEliece Cryptosystem Implementation: Theory and Practice. In: Buchmann, J., Ding, J. (eds.) PQCrypto 2008. LNCS, vol. 5299, pp. 47–62. Springer, Heidelberg (2008)
12. Bresson, E., Stern, J., Szydlo, M.: Threshold Ring Signatures and Applications to Ad-hoc Groups. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 465–480. Springer, Heidelberg (2002)
13. Cayrel, P.-L., Lindner, R., Rückert, M., Silva, R.: A Lattice-Based Threshold Ring Signature Scheme. In: Abdalla, M., Barreto, P.S.L.M. (eds.) LATINCRYPT 2010. LNCS, vol. 6212, pp. 255–272. Springer, Heidelberg (2010)
14. Cayrel, P.-L., Véron, P., El Yousfi Alaoui, S.M.: A Zero-Knowledge Identification Scheme Based on the q-ary Syndrome Decoding Problem. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) SAC 2010. LNCS, vol. 6544, pp. 171–186. Springer, Heidelberg (2011)
15. Courtois, N.T., Finiasz, M., Sendrier, N.: How to Achieve a McEliece-Based Digital Signature Scheme. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 157–174. Springer, Heidelberg (2001)
16. Dallot, L., Vergnaud, D.: Provably Secure Code-Based Threshold Ring Signatures. In: Parker, M.G. (ed.) Cryptography and Coding 2009. LNCS, vol. 5921, pp. 222–235. Springer, Heidelberg (2009)
17. Fiat, A., Shamir, A.: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)
18. Fiege, U., Fiat, A., Shamir, A.: Zero knowledge proofs of identity. In: Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing - STOC, pp. 210–217 (1987)
19. Gaborit, P., Girault, M.: Lightweight code-based authentication and signature. In: IEEE International Symposium on Information Theory–ISIT 2007, Nice, France, pp. 191–195. IEEE (2007)
20. Herranz, J., Sáez, G.: Forking Lemmas for Ring Signature Schemes. In: Johansson, T., Maitra, S. (eds.) INDOCRYPT 2003. LNCS, vol. 2904, pp. 266–279. Springer, Heidelberg (2003)
21. May, A., Meurer, A., Thomae, E.: Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 107–124. Springer, Heidelberg (2011)
22. McEliece, R.: A public-key cryptosystem based on algebraic coding theory. The Deep Space Network Progress Report, DSN PR 42–44 (1978), `http://ipnpr.jpl.nasa.gov/progressreport2/42-44/44N.PDF`

23. Niebuhr, R., Cayrel, P.-L., Bulygin, S., Buchmann, J.: On Lower Bounds for Information Set Decoding over $\mathbb{F}_q$. In: SCC 2010, RHUL, London, UK, pp. 143–157 (2010)
24. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Problems of Control and Information Theory 15(2), 159–166 (1986)
25. Peters, C.: Information-Set Decoding for Linear Codes over $\mathbb{F}_q$. In: Sendrier, N. (ed.) PQCrypto 2010. LNCS, vol. 6061, pp. 81–94. Springer, Heidelberg (2010)
26. Rivest, R.L., Shamir, A., Tauman, Y.: How to Leak a Secret: Theory and Applications of Ring Signatures. In: Goldreich, O., Rosenberg, A.L., Selman, A.L. (eds.) Theoretical Computer Science. LNCS, vol. 3895, pp. 164–186. Springer, Heidelberg (2006)
27. Stern, J.: A New Identification Scheme Based on Syndrome Decoding. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 13–21. Springer, Heidelberg (1994)
28. Stern, J.: A new paradigm for public key identification. IEEE Transactions on Information Theory 42, 1757–1768 (1996)

A Encoding Function over $\mathbb{F}_q$

The constant weight encoding bijective function φ_q is described in Algorithm 2, this function takes its input from the interval $[0, (q-1)^{\omega}\binom{n}{\omega}[$ and outputs a q-ary word of length n and Hamming weight ω. Algorithm 2 uses a binary encoder method introduced by Biswas and Sendrier [11]. This function is a constant weight encoding function taking $s = \omega \log_2(n/\omega)$ input bits and outputing a binary word of length n and weight ω, and which is very efficient because of its linear time encoding.

Algorithm 2. q-ary EnumDecoding

Input: integers n, q, ω and $x \in [0, (q-1)^{\omega}\binom{n}{\omega}[, (\omega \leq n)$
Output: q-ary word of length n and Hamming weight ω

1: $db \leftarrow \lfloor x/(q-1)^{\omega} \rfloor$
2: ret $\leftarrow$ binary encoder(db, n, ω)
3: $rest \leftarrow x$ **mod** $(q-1)^{\omega}$
4: **for** i from 1 to n **do**
5: if $0 <$ ret$[i]$
6: ret$[i] \leftarrow (rest$ **mod** $(q-1)) + 1$
7: $rest \leftarrow \lfloor rest/(q-1) \rfloor$
8: end if
9: end **for**
10: **Return** ret